AF
IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Steven J. Sistare, Nicholas J. Nevin and Anthony L. Kimball §

Group Art Unit: 2194 §

Attorney Docket: 5181-93900 §

Serial No.: 09/303,464 §

Filed: April 30, 1999 §

For: System and Method for Controlling Co-Scheduling Of Processes Of Parallel Programs §

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, Box Non-Fee Amendment, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below:

B. Noël Kivlin
Registered Representative

7-27-05
Date

Signature

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed June 2, 2005, Appellants present this Appeal Brief. Appellants respectfully request that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The present application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having an office and place of business at a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 901 San Antonio Road, Palo Alto, California 94303.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to Appellants, Appellants' legal representatives, or assignee which will directly affect, be directly affected by, or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1 – 15 are pending. Claims 1 – 15 are rejected, and the rejection of these claims is being appealed. A copy of claims 1 – 15 is included in the Claims Appendix attached hereto.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a system for controlling co-scheduling of processes in a computer comprising at least one process (FIG. 1, reference numerals 12(1)(1) – 12(1)(M_1), 12(N)(1) – 12 (N,M_N)) and a spin daemon (FIG. 1, reference numeral 17). The at least one process is configured to, when it is waiting for a flag (FIG. 1, reference numeral 16(1)(1) – 16(N)(M_N)) to change condition, transmit a flag monitor request to the spin daemon (FIG. 2A, reference numeral 112; page 6, lines 7 – 11; page 10, lines 18 - 21) and de-schedule itself (FIG. 2A, reference numeral 113; page 6, lines 10 – 11; page 10, line 21). After receiving the flag monitor request (FIG. 2A, reference numeral 114; page 6, lines 7 – 10; page 10, line 22), the spin daemon is configured to monitor the flag (FIG. 2B, reference numeral 117; page 6, lines 7 – 10; page 6, lines 20 – 21; page 11, lines 1 - 4). After the flag changes condition (page 6, line 22; page 11, lines 8 - 10), the spin daemon is configured to enable the at least one process to be re-scheduled for execution by the computer (FIG. 2B, reference numerals 121, 122 and FIG. 2C, reference numerals 123, 124; page 6, lines 22 – 25; page 11, lines 11 - 20).

Independent claim 6 is directed to a method of controlling co-scheduling of processes in a computer comprising at least one process (FIG. 1, reference numerals 12(1)(1) – 12(1)(M_1), 12(N)(1) – 12 (N,M_N)) and a spin daemon (FIG. 1, reference numeral 17). The method comprises the steps of (A) enabling the at least one process to, when it is waiting for a flag (FIG. 1, reference numeral 16(1)(1) – 16(N)(M_N)) to change condition, transmit a flag monitor request to the spin daemon (FIG. 2A, reference numeral 112; page 6, lines 7 – 11; page 10, lines 18 - 21) and de-schedule itself (FIG. 2A, reference numeral 113; page 6, lines 10 – 11; page 10, line 21) and (B) enabling the spin daemon to, after receiving the flag monitor request (FIG. 2A, reference numeral 114; page 6, lines 7 – 10; page 10, line 22), monitor the flag (FIG. 2B, reference numeral 117; page 6, lines 7 – 10; page 6, lines 20 – 21; page 11, lines 1 - 4) and, after the flag changes condition (page 6, line 22; page 11, lines 8 - 10), enable the at least one process to be re-scheduled for execution by the computer (FIG. 2B, reference numerals 121, 122 and FIG. 2C, reference numerals 123, 124; page 6, lines 22 – 25; page 11, lines 11 - 20).

Independent claim 11 is directed to a computer program product for use in connection with a computer to control co-scheduling of at least one process (FIG. 1, reference numerals 12(1)(1) – 12(1)(M_1), 12(N)(1) – 12 (N,M_N)) in the computer, the computer program product including a computer readable medium having encoded thereon a process module and a spin daemon module. The process module is configured to enable the computer to, when the at least one process is waiting for a flag (FIG. 1, reference numeral 16(1)(1) – 16(N)(M_N)) to change condition, transmit a flag monitor request (FIG. 2A, reference numeral 112; page 6, lines 7 – 11; page 10, lines 18 - 21) and de-schedule itself (FIG. 2A, reference numeral 113; page 6, lines 10 – 11; page 10, line 21). The spin daemon module is configured to enable the computer to, after receiving the flag monitor request (FIG. 2A, reference numeral 114; page 6, lines 7 – 10; page 10, line 22), monitor the flag (FIG. 2B, reference numeral 117; page 6, lines 7 – 10; page 6, lines 20 – 21; page 11, lines 1 - 4) and, after the flag changes condition (page 6, line 22; page 11, lines 8 - 10), enable the at least one process to be re-scheduled for execution by the computer (FIG. 2B, reference numerals 121, 122 and FIG. 2C, reference numerals 123, 124; page 6, lines 22 – 25; page 11, lines 11 - 20).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 2, 6, 7, 11 and 12 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Inakoshi (U.S. Patent No. 5,933,604, hereinafter "Inakoshi").

2. Claims 3, 4, 8, 9, 13 and 14 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Inakoshi in view of Conger (Windows API Bible, 1992 Publication).

3. Claims 5, 10 and 15 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Inakoshi in view of Arthur Dumas (Programming WinSock, 1995 Publication, hereinafter "Dumas").

VII. ARGUMENT

First Ground of Rejection:

Claims 1, 2, 6, 7, 11 and 12 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Inakoshi (U.S. Patent No. 5,933,604, hereinafter "Inakoshi"). Appellants traverse this rejection for the following reasons.

Claims 1 and 2:

Appellants respectfully submit that Inakoshi does not teach or suggest a system comprising at least one process and a spin daemon, wherein the at least one process is configured in pertinent part, to "**de-schedule itself**" when it is waiting for a flag to change condition, and wherein the spin daemon is configured to "**enable the at least one process to be re-scheduled for execution**" after the flag changes condition, as recited in Appellants' Claim 1.

Inakoshi is directed to a network resource monitoring system and method (Abstract). There is no mention of scheduling, "de-scheduling" or "re-scheduling" processes anywhere in Inakoshi. The Final Office Action acknowledges that Inakoshi does not teach that the process is configured to de-schedule itself. However, in rejecting Claim 1, the Final Office Action asserts that Inakoshi teaches that "by sending the monitor request to a monitor system, the user does not have to access the resource by itself, the change in the state of the resource will be automatically noticed to the user by the monitor system (lines 26 – 35 column 5)" (Page 3, paragraph 1 and Page 6, paragraph 2 of the Final Office Action) and that "therefore, one of ordinary skill in the art would conclude that in this case, the process from the user has been de-schedule itself from accessing the resource by itself". Appellants respectfully disagree. More particularly, the alleged teaching in Inakoshi that "a user does not have to access a resource by itself, the change in the state of the resource will be automatically noticed to the user by the monitor system", does not lead to a conclusion that any process anywhere "de-schedules itself" as recited in Claim 1.

Further with respect to Claim 1, the Final Office Action asserts that the feature of the spin daemon configured to enable the at least one process to be re-scheduled by the computer after the flag changes condition is taught by Inaksohi, and cites col. 4, lines 31 – 37 and col. 4, lines 30 and 40 of Inakoshi in support of this assertion. Appellants respectfully disagree with this assertion. In lines 29 – 46 of col. 4, Inakoshi discloses:

> "The output unit 2 outputs change information that indicates a change in the state of the resource 6.
>
> The resource 6 is, for example, image information and/or text information, such as a home page on the Internet, that is created and transmitted by a user. Which resource on the communication network 5 is the target of monitoring is specified at the time of the user request. The monitoring request from the user is sent, for example, from the management unit 4 to the monitoring unit 1.
>
> The monitoring unit 1 for example accesses the resource 6 and checks on its state periodically. When it is determined that the resource has been updated, the monitoring unit 1 transmits the fact that the state of the resource 6 has changed to the output unit 2. Upon receiving this notice, the output unit 2 outputs change information, including information that identifies the resource 6, to the communication network 5."

Appellants respectfully submit that Inakoshi does not teach or suggest a spin daemon enabling any process to be re-scheduled after a flag changes condition, in the cited lines or anywhere else.

In addition, neither Conger nor Dumas, taken singly or in combination with Inakoshi, teach or suggest the combination of features recited in Claim 1. Accordingly, Claim 1 along with its dependent Claim 2 are believed to patentably distinguish over the art cited in the Final Office Action.

Claims 6, 7, 11 and 12:

In language similar to that of Claim 1, Claim 6 recites a method comprising, in pertinent part, enabling at least one process to, when it is waiting for a flag to change condition, **de-schedule itself**, and enabling a spin daemon to **enable the process to be re-scheduled after the flag**

changes condition. Similarly, Claim 11 recites, in pertinent part, a process module configured to enable a computer to **de-schedule itself,** and a spin daemon module configured to enable the computer to **enable the at least one process to be re-scheduled** for execution by the computer. For at least the reasons cited above with respect to Claim 1, Claims 6 and 11, along with their respective dependent claims 7 and 12, are believed to patentably distinguish over Inakoshi.

Second Ground of Rejection:

Claims 3, 4, 8, 9, 13 and 14 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Inakoshi in view of Conger (Windows API Bible, 1992 Publication). Appellants traverse this rejection for the following reasons.

Claims 3, 8 and 13:

Claims 3 depends upon Claim 2, and is therefore believed to be patentable over Inakoshi and Conger for the reasons cited above in Appellants' arguments with respect to the first ground of rejection.

In addition, Claim 3 recites a system in which **a plurality of flags is contained in a memory segment**, and in which the spin daemon is configured to enable the at least one process to be re-scheduled following a change of condition of **any flag in said memory segment.** In rejecting Claim 3, the Final Office Action asserted that, because Conger "teaches memory segment", "it would have been obvious to apply the teachings of Conger to the system of Inakoshi because this provides the flags of Inakoshi are contained in a memory segment, therefore the memory holding the flags can be deallocated when that process is finished. As a result, the system of Inakoshi will work more efficiently" (Page 4, Section 4 of the Final Office Action). Appellants respectfully disagree. The Final Office Action does not appear to address the limitation of the spin daemon enabling a process to be rescheduled following a change of **any flag of a plurality of flags** contained in a memory segment. Applicants respectfully submit that neither Conger, Inakoshi, nor Dumas, taken singly or in combination, teach or suggest such a

spin daemon, and that the rejection of Claim 3 is therefore in error.

Claims 8 and 13 depend respectively upon Claims 7 and 12, and are therefore also believed to be patentable over Inakoshi and Conger for the reasons cited above in Appellants' arguments with respect to the first ground of rejection. In addition, Claims 8 and 13 also recite limitations similar to those of Claim 3, and are therefore believed to further distinguish over the art cited by the Final Office Action for at least the reasons cited above in Appellants' arguments with respect to Claim 3.

Claims 4, 9 and 14:

Claims 4 depends upon Claim 3, and is therefore believed to be patentable over Inakoshi and Conger for the reasons cited above in Appellants' arguments with respect to Claim 3.

In addition, Claim 4 recites that the at least one process is configured to "**register with said spin daemon, during registration** the at least one process being configured to provide the spin daemon with an identifier for the memory segment, the spin daemon being configured to provide a handle, the at least one process being configured to use the handle in the flag monitor request". In rejecting Claim 4, the Final Office Action does not appear to address the limitation of a process **registering** with a spin daemon and providing an identifier for the memory segment **during registration**. Furthermore, the Final Office Action cites various portions of Conger that "provide an identifier for the memory segment" and "the use of a handle", and asserts that "it would have been obvious to provide a handle in Inakoshi's process (a spin daemon) so that the handle can be used to perform the flag monitor request by any other processes of Inakoshi's system, and the handles in Conger can trace different types of system resources" (Page 4, Section 4 of the Final Office Action). Appellants respectfully disagree. The alleged teachings of Conger on "the use of a handle" and "providing an identifier for a memory segment", when combined with the teachings of Inakoshi, do not lead to a process registering with a spin daemon, providing an identifier of a memory segment containing a plurality of flags to the spin daemon during

registration, and using a handle provided by the spin daemon in a flag monitor request, as recited in Claim 4. In addition, the Final Office Action asserts that the motivation for combining the teachings of Conger and Inakoshi is "so that the handle can be used to perform the flag monitor request by any other processes of Inakoshi's system, and the handles in Conger can trace different types of system resources". Appellants respectfully submit that the motivation provided by the Final Office Action is unsupported by the cited art; Appellants can find no teaching or suggestion in the cited art of how or why a handle provided to one process should be "used to perform the flag monitor request by any other processes" or "to trace different types of system resources". Appellants respectfully assert that neither Inakoshi, Conger, nor Dumas, taken singly or in combination, teach or suggest the combination of limitations recited in Claim 4. Claim 4 is therefore believed to further distinguish over the art cited in the Final Office Action.

Claims 9 and 14 depend respectively upon Claims 8 and 13, and are therefore also believed to be patentable over Inakoshi and Conger for the reasons cited above in Appellants' arguments with respect to Claims 8 and 13. In addition, Claims 9 and 14 also recite limitations similar to those of Claim 4, and are therefore believed to further distinguish over the art cited by the Final Office Action for at least the reasons cited above in Appellants' arguments with respect to Claim 4.

Third Ground of Rejection:

Claims 5, 10 and 15 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Inakoshi in view of Arthur Dumas (Programming WinSock, 1995 Publication).

Claims 5, 10 and 15 depend upon independent claims 1, 6 and 11, and are therefore believed to patentably distinguish over the art cited in the Final Office Action for at least the reasons given above in Appellants' arguments with respect to the first ground of rejection.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1 - 15 was erroneous, and reversal of the decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 50-1505/5181-93900/BNK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

B. Noël Kivlin
Reg. No. 33,929
ATTORNEY FOR APPELLANT(S)

Meyertons, Hood, Kivlin, Kowert and Goetzel, P.C.
P.O. Box 398
Austin, Texas 78767-0398
Phone: (512) 853-8800
Date: 7-27-05

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A system for controlling co-scheduling of processes in a computer comprising at least one process and a spin daemon, the at least one process being configured to, when it is waiting for a flag to change condition, transmit a flag monitor request to the spin daemon and de-schedule itself, the spin daemon being configured to, after receiving the flag monitor request monitor the flag and, after the flag changes condition, enable the at least one process to be re-scheduled for execution by the computer.

2. A system as defined in claim 1 in which said spin daemon is configured to monitor a plurality of flags, each in response to a flag monitor request, the spin daemon maintaining a list identifying those flags it is to monitor, the spin daemon being further configured to, when it receives a flag monitor request, add an identification of a flag associated with the request to the list.

3. A system as defined in claim 2 in which said flags are contained in a memory segment, the spin daemon being configured to enable the at least one process to be re-scheduled following a change of condition of any flag in said memory segment.

4. A system as defined in claim 3 in which said at least one process is configured to register with said spin daemon, during registration the at least one process being configured to provide the spin daemon with an identifier for the memory segment, the spin daemon being configured to provide a handle, the at least one process being configured to use the handle in the flag monitor request.

5. A system as defined in claim 1 in which said at least one process and said spin daemon are configured to communicate over a socket.

6. A method of controlling co-scheduling of processes in a computer comprising at least one process and a spin daemon, the method comprising the steps of:

A. enabling the at least one process to, when it is waiting for a flag to change condition, transmit a flag monitor request to the spin daemon and de-schedule itself,

B. enabling the spin daemon to, after receiving the flag monitor request monitor the flag and, after the flag changes condition, enable the at least one process to be re-scheduled for execution by the computer.

7. A method as defined in claim 6, the spin daemon being configured to monitor a plurality of flags, each in response to a flag monitor request, the spin daemon maintaining a list identifying those flags it is to monitor, the method including the step of enabling the spin daemon being to, when it receives a flag monitor request, add an identification of a flag associated with the request to the list.

8. A method as defined in claim 7 in which said flags are contained in a memory segment, the method including the step of enabling the spin daemon to enable the at least one process to be re-scheduled following a change of condition of any flag in said memory segment.

9. A method as defined in claim 8 further including the steps of

A enabling the at least one process to register with said spin daemon, during registration the at least one process being configured to provide the spin daemon with an identifier for the memory segment; and

B. enabling the spin daemon to provide a handle for use by the at least one process in the flag monitor request.

10. A method as defined in claim 6 further comprising the step of enabling the at least one process and said spin daemon to communicate over a socket.

11. A computer program product for use in connection with a computer to control co-scheduling of at least one process in the computer, the computer program product including a computer readable medium having encoded thereon:

A. a process module configured to enable the computer to, when the at least one process is waiting for a flag to change condition, transmit a flag monitor request and de-schedule itself,

B. a spin daemon module configured to enable the computer to, after receiving the flag monitor request, monitor the flag and, after the flag changes condition, enable the at least one process to be re-scheduled for execution by the computer.

12. A computer program product as defined in claim 11 in which said spin daemon is configured to enable the computer to monitor a plurality of flags, each in response to a flag monitor request, the spin daemon enabling the computer to maintain a list identifying those flags it is to monitor, the spin daemon being further configured to enable the computer to, when it receives a flag monitor request, add an identification of a flag associated with the request to the list.

13. A computer program product as defined in claim 12 in which said flags are contained in a memory segment, the spin daemon being configured to enable the computer enable the at least one process to be re-scheduled following a change of condition of any flag in said memory segment.

14. A computer program product as defined in claim 13 in which said at least one process is configured to enable the computer to register with said spin daemon, during registration the at least one process being configured to enable the computer to provide the spin daemon with an identifier for the memory segment, the spin daemon being configured to enable the computer to provide a handle, the at least one process being configured to use the handle in the flag monitor request.

15. A computer program product as defined in claim 11 in which said at least one process and said spin daemon are configured to enable the computer to communicate over a socket.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings known to Appellants, Appellants' legal representatives, or assignee which will directly affect, be directly affected by, or have a bearing on the Board's decision in the pending appeal.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	§	Group Art Unit: 2194
	§	Examiner: Andy Ho
Sistare, et al.	§	
	§	Atty. Dkt.: 5181-93900
Serial No.: 09/303,464	§	I hereby certify that this correspondence is being deposited with the U.S. Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below:
Filed: April 30, 1999	§	B. Noël Kivlin Registered Representative
For: SYSTEM AND METHOD FOR CONTROLLING CO-SCHEDULING OF PROCESSES OF PARALLEL PROGRAMS	§	July 27, 2005 (Date) _____ (Signature)

FEE AUTHORIZATION

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

The Commissioner is hereby authorized to charge the following fee to Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C. Deposit Account Number 50-1505/5181-93900:

Fee:	Appeal Brief
Amount:	$500.00
Attorney Docket No.:	5181-93900

The Commissioner is also authorized to charge any extension fee or other fees which may be necessary to the same account number.

Respectfully submitted,

B. Noël Kivlin
Reg. No. 33,929
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin,
Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Ph: (512) 853-8800
Date: July 27, 2005